

F4–16AD–2

8-Channel

Analog Input

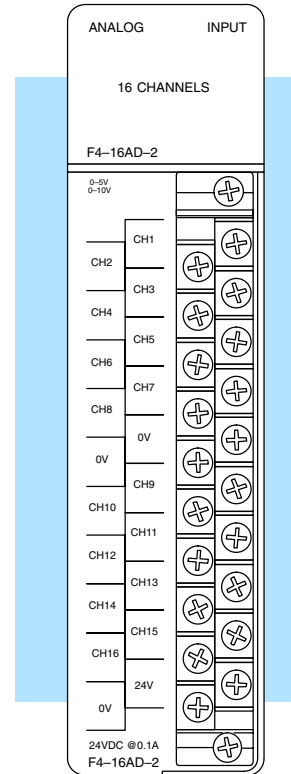
In This Chapter. . . .

- Module Specifications
 - Setting the Module Jumpers
 - Connecting the Field Wiring
 - Module Operation
 - Writing the Control Program
-

Module Specifications

The F4-16AD-2 Analog Input module provides several features and benefits.

- It accepts 16 single-ended voltage inputs.
- Analog inputs are optically isolated from PLC logic components.
- The module has a removable terminal block, so the module can be easily removed or changed without disconnecting the wiring.
- All 16 analog inputs may be read in one CPU scan (D4-440 and D4-450 CPUs only).



Analog Input Configuration Requirements

The F4-16AD-2 Analog Input module requires 16 discrete input points. The module can be installed in any slot of a DL405 system, including remote bases. The limitations on the number of analog modules are:

- For local and expansion systems, the available power budget and discrete I/O points.
- For remote I/O systems, the available power budget and number of remote I/O points.

Check the user manual for your particular model of CPU for more information regarding power budget and number of local or remote I/O points.

The following table provides the specifications for the F4-16AD-2 Analog Input Module. Review these specifications to ensure the module meets your application requirements.

Input Specifications

Number of Channels	16, single ended (one common)
Input Ranges	0-5V, 0-10V
Resolution	12 bit (1 in 4096)
Active Low-pass Filtering	-3 dB at 20Hz, -6 dB per octave
Input Impedance	100K Ohm minimum
Absolute Maximum Ratings	130VAC / 100VDC
Conversion Time	0.4ms per channel (module conversion) 2 ms per selected channel minimum (CPU)
Linearity Error (End to End)	± 2 count (0.050% of full scale) maximum
Input Stability	± 1 count
Full Scale Calibration Error (Offset error not included)	± 12 counts maximum , voltage input
Offset Calibration Error	± 3 counts maximum, unipolar voltage input

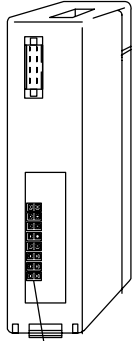
General Specifications

PLC Update Rate	16 Channel per scan max.
Digital Input Points Required	16 (X) input points total 12 binary data bits, 4 active channel bits,
Power Budget Requirement	75 mA (power from base)
External Power Supply	21.6-26.4 VDC, 100 mA, class 2
Accuracy vs. Temperature	± 50 ppm / °C maximum full scale (including maximum offset change of 2 counts)
Operating Temperature	0 to 60°C (32 to 140° F)
Storage Temperature	-20 to 70°C (-4° F to 158° F)
Relative Humidity	5 to 95% (non-condensing)
Environmental Air	No corrosive gases permitted
Vibration	MIL STD 810C 514.2
Shock	MIL STD 810C 516.2
Noise Immunity	NEMA ICS3-304

One count in the specification table is equal to one least significant bit of the analog data (1 in 4096).

Setting the Module Jumpers

Jumper Locations



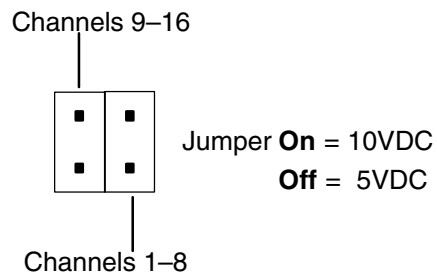
Jumper Locations

If you examine the rear of the module, you will notice two banks of jumpers. The module has several options that you can select by installing or removing these jumpers:

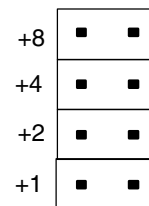
- A bank of two jumpers to set voltage input range for each bank of 8 channels.
- A bank of four jumpers to select the number of channels used.

The module is set at the factory for a 0–10 VDC signal range on all sixteen channels. The following diagram shows how the jumpers are set at the factory and describes the function of each jumper. When removing a jumper, store it by placing it on a single pin to prevent losing it.

Voltage Input Range Select



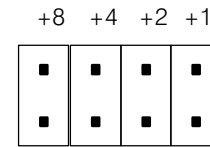
Number of Channels



Selecting the Number of Channels

The jumpers labeled +1, +2, +4 and +8 are used to select the number of channels that will be used.

Any unused channels are not processed so if you only select channels 1-8, then the last eight channels will not be active. The following table shows which jumpers to install.




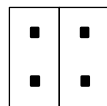
Number of Channels

Jumpers installed as shown selects 16-channel operation

Channel(s)	Jumper				Channel(s)	Jumper			
	+8	+4	+2	+1		+8	+4	+2	+1
1	No	No	No	No	1 2 3 4 5 6 7 8 9	Yes	No	No	No
1 2	No	No	No	Yes	1 2 3 4 5 6 7 8 9 10	Yes	No	No	Yes
1 2 3	No	No	Yes	No	1 2 3 4 5 6 7 8 9 10 11	Yes	No	Yes	No
1 2 3 4	No	No	Yes	Yes	1 2 3 4 5 6 7 8 9 10 11 12	Yes	No	Yes	Yes
1 2 3 4 5	No	Yes	No	No	1 2 3 4 5 6 7 8 9 10 11 12 13	Yes	Yes	No	No
1 2 3 4 5 6	No	Yes	No	Yes	1 2 3 4 5 6 7 8 9 10 11 12 13 14	Yes	Yes	No	Yes
1 2 3 4 5 6 7	No	Yes	Yes	No	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	Yes	Yes	Yes	No
1 2 3 4 5 6 7 8	No	Yes	Yes	Yes	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16	Yes	Yes	Yes	Yes

Selecting Input Signal Ranges

The following table shows the jumper selections for the 5V and 10V ranges. The module comes from the factory set for the 10V operation (jumpers installed). See note below.

Signal Range	Jumper Settings
0 to +5 VDC Jumpers OFF	Channels 1-8 Jumper  Channels 9-16 Jumper
0 to +10 VDC Jumpers ON	Channels 1-8 Jumper  Channels 9-16 Jumper



NOTE: The jumpers do not have to both be set On or Off (e.g. Channels 1-8 can be selected for 5V and Channels 9-16 can be selected for 10V operation).

Connecting the Field Wiring

Wiring Guidelines

Your company may have guidelines for wiring and cable installation. If so, you should check those before you begin the installation. Here are some general things to consider.

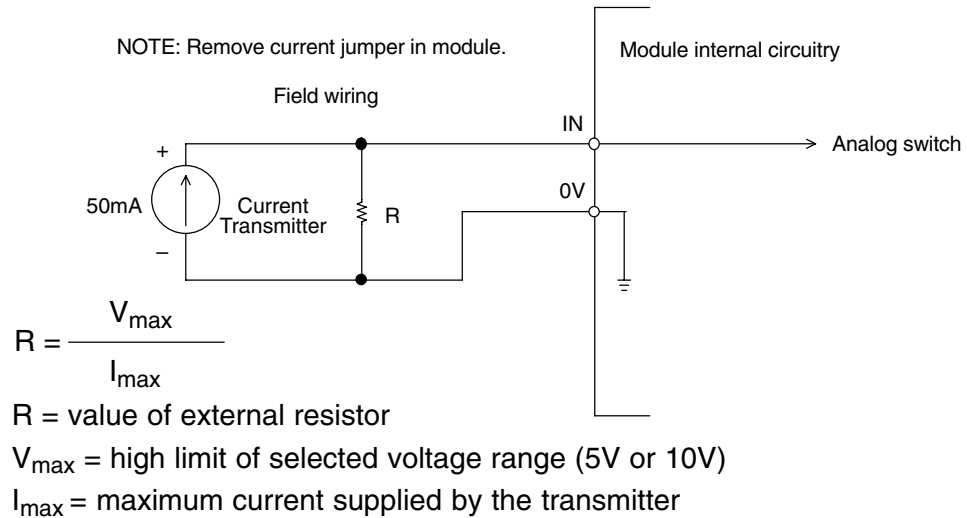
- Use the shortest wiring route whenever possible.
- Use shielded wiring and ground the shield at the transmitter source. *Do not* ground the shield at both the module and the source.
- Don't run the signal wiring next to large motors, high current switches, or transformers. This may cause noise problems.
- Route the wiring through an approved cable housing to minimize the risk of accidental damage. Check local and national codes to choose the correct method for your application.

User Power Supply Requirements

The F4-16AD-2 module requires a separate power supply. The Series DL405 CPUs, D4-RS Remote I/O Controller, and D4-EX Expansion Units have built-in 24 VDC power supplies that provide up to 400mA of current. If you only have a couple analog modules, you can use this power source instead of a separate supply. If you have more than four analog modules, or you would rather use a separate supply, choose one that meets the following requirements: 24 VDC \pm 10%, Class 2, 100 mA current (per module).

Custom Input Ranges

Occasionally you may have the need to connect a transmitter with an unusual signal range. By changing the wiring slightly and adding an external resistor to convert the current to voltage, you can easily adapt this module to meet the specifications for a transmitter that does not adhere to one of the standard input ranges. The following diagram shows how this works.



Example: current transmitter capable of 50mA, 0 – 10V range selected.

$$R = \frac{10V}{50mA} \quad R = 200 \text{ ohms}$$



NOTE: Your choice of resistor can affect the accuracy of the module. A resistor that has $\pm 0.1\%$ tolerance and a $\pm 50\text{ppm} / ^\circ\text{C}$ temperature coefficient is recommended.

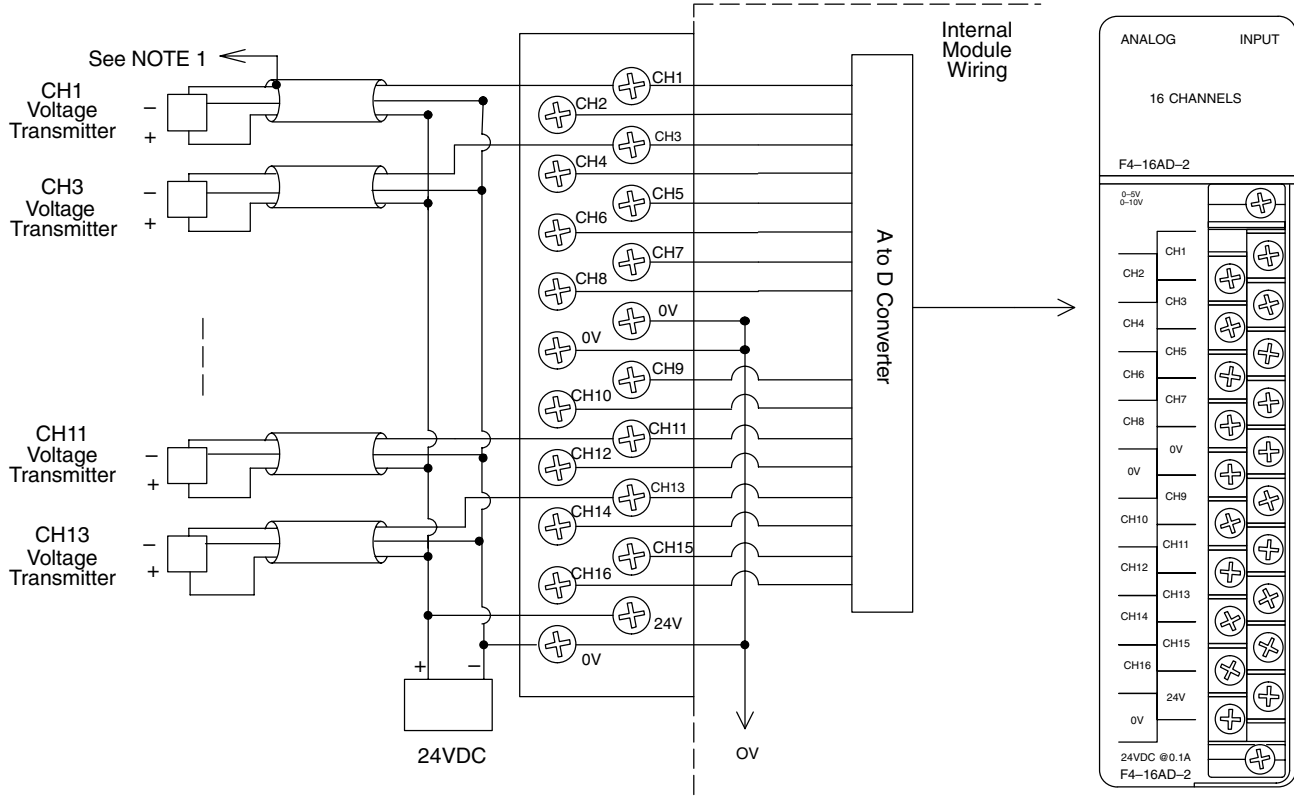
Removable Connector

The F4-16AD-2 module has a removable connector to make wiring easier. Simply remove the retaining screws and gently pull the connector from the module.

Wiring Diagram

NOTE 1: Shields should be grounded at the signal source.

F4-16AD-2
16-Channel Analog Input



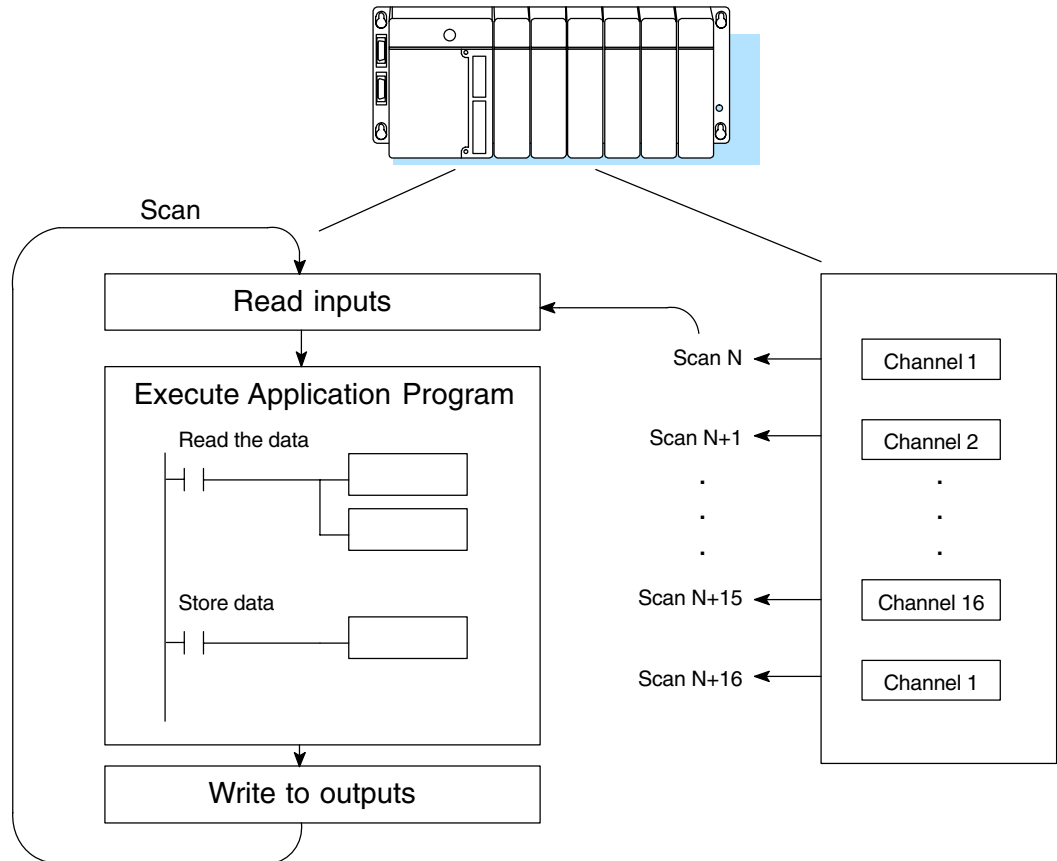
If the power supply common of an external power supply is not connected to 0V on the module, then the output of the external transmitter must be isolated.

Channel Scanning Sequence

Before you begin writing the control program, it is important to take a few minutes to understand how the module processes and represents the analog signals.

The F4-16AD-2 module supplies one channel of data per each CPU scan. Since there are sixteen channels, it can take up to sixteen scans to get data for all channels. Once all channels have been scanned the process starts over with channel 1. There are ways around this. Later we'll show you how to write a program that will get all sixteen channels in one scan.

Unused channels are not processed, so if you select only two channels, then each channel will be updated every other scan.



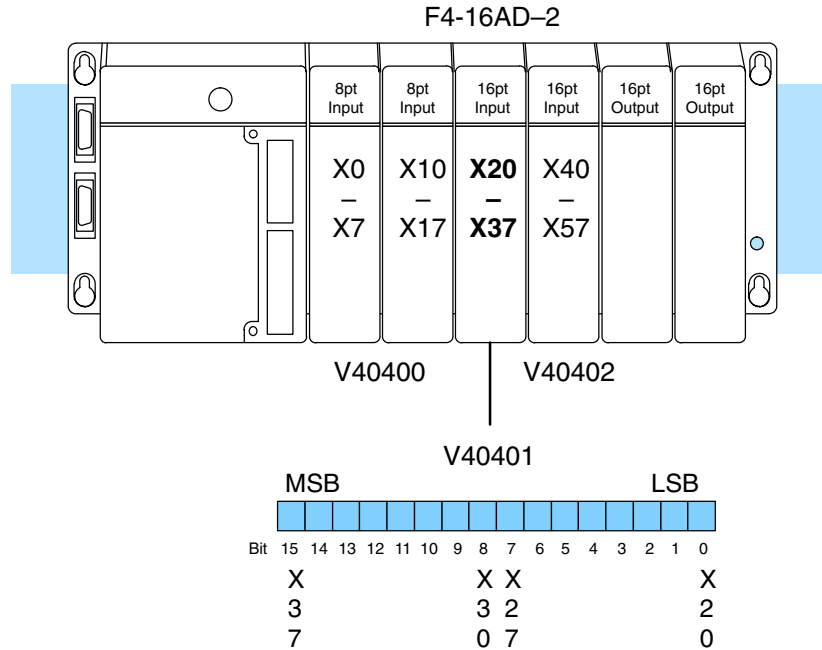
Even though the channel updates to the CPU are synchronous with the CPU scan, the module asynchronously monitors the analog transmitter signal and converts the signal to a 12-bit binary representation. This enables the module to continuously provide accurate measurements without slowing down the discrete control logic in the RLL program.

Input Bit Assignments

You may recall the F4-16AD-2 module requires 16 discrete input points from the CPU. These 16 points provide:

- An indication of which channel is active.
- The digital representation of the analog signal.

Since all input points are automatically mapped into V-memory, it is very easy to determine the location of the data word that will be assigned to the module.

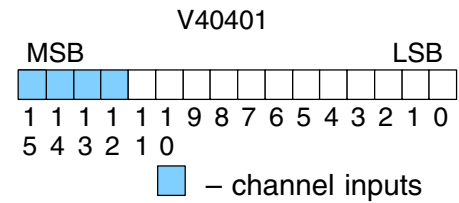


F4-16AD-2
16-Channel Analog Input

Within this word location, the individual bits represent specific information about the analog signal.

Active Channel Indicator Inputs

The bits (inputs) shown in the diagram indicate the active channel. The next to last four bits of the V-memory location indicate the active channel. The inputs are automatically turned on and off on each CPU scan to indicate the active channel.

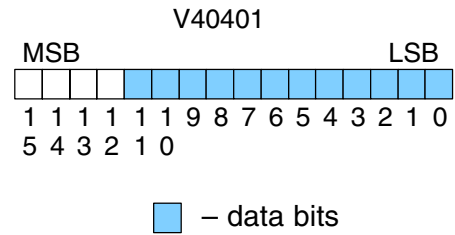


Scan	Channel Inputs	Channel	Scan	Channel Inputs	Channel
N	0000	1	N+8	1000	9
N+1	0001	2	N+9	1001	10
N+2	0010	3	N+10	1010	11
N+3	0011	4	N+11	1011	12
N+4	0100	5	N+12	1100	13
N+5	0101	6	N+13	1101	14
N+6	0110	7	N+14	1110	15
N+7	0111	8	N+15	1111	16

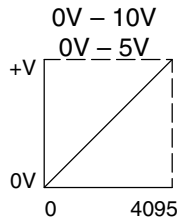
Analog Data Bits

The first twelve bits represent the analog data in binary format.

Bit	Value	Bit	Value
0	1	6	64
1	2	7	128
2	4	8	256
3	8	9	512
4	16	10	1024
5	32	11	2048



Since the module has 12-bit resolution, the analog signal is converted into 4096 counts ranging from 0-4095 (2^{12}). For example, with a 0 to 10V scale, a 0V signal would be 0, and a 10V signal would be 4095. This is equivalent to a binary value of 0000 0000 0000 to 1111 1111 1111, or 000 to FFF hexadecimal. The following diagram shows how this relates to each signal range.



Each count can also be expressed in terms of the signal level by using the equation shown. The following table shows the smallest signal levels that will result in a change in the data value for each signal range.

$$\text{Resolution} = \frac{H - L}{4095}$$

H = high limit of the signal range

L = low limit of the signal range

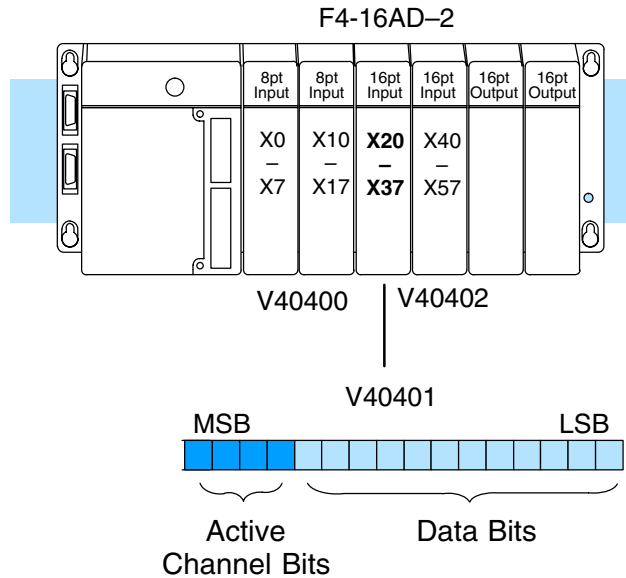
Range	Signal Span (H - L)	Divide By	Smallest Detectable Change
0 to 5V	5 V	4095	1.22 mV
0 to 10V	10 V	4095	2.44 mV

Writing the Control Program

If you have configured the F4-16AD-2 module, use the following examples to get started writing the control program.

Multiple Channels Selected

Since all channels are multiplexed into a single data word, the control program must be set up to determine which channel is being read. Since the module appears as X input points to the CPU, it is very easy to use the active channel status bits to determine which channel is being monitored.



F4-16AD-2
16-Channel Analog Input

Storing the Channel Data to V-Memory

The OUTX instruction used the following programming examples that follow stores the channel data to an address that starts at V3000 plus the channel offset. For example, if channel 2 was being read, the data would be stored in V3002 (V3000 + 2).

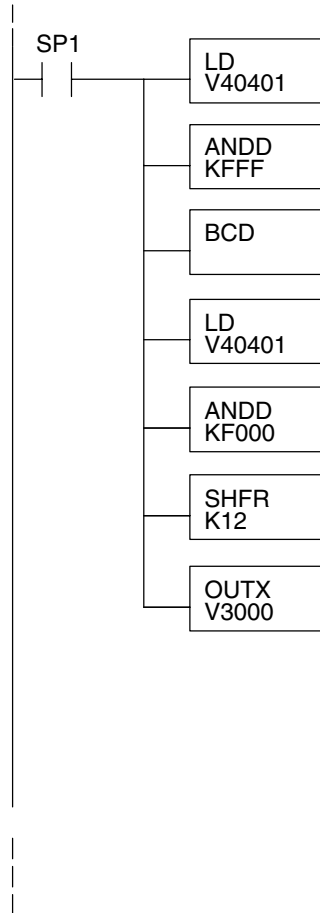
Module Reading	Acc. Bits	Offset	Data Stored in ...
Channel 1	0000	0	V3000
Channel 2	0001	1	V3001
Channel 3	0010	2	V3002
Channel 4	0011	3	V3003
Channel 5	0100	4	V3004
Channel 6	0101	5	V3005
Channel 7	0110	6	V3006
Channel 8	0111	7	V3007
Channel 9	1000	8	V3010
Channel 10	1001	9	V3011
Channel 11	1010	10	V3012
Channel 12	1011	11	V3013
Channel 13	1100	12	V3014
Channel 14	1101	13	V3015
Channel 15	1110	14	V3016
Channel 16	1111	15	V3017

Reading Values



430 440 450

The following program example shows how to read the analog data into V-memory locations with the D4-430 CPU. Since the D4-430 does not support the LDF instruction, you can use the LD instruction instead as shown. The example also works for D4-440 and D4-450 CPUs. This example will read one channel per scan, so it will take sixteen scans to read all sixteen channels. Contact SP1 is used in the example because the inputs are continually being updated.



Loads the complete channel data word from the module into the accumulator. The V-memory location depends on the I/O configuration. See Appendix A for the memory map.

This instruction masks the channel identification bits. Without this, the values used will not be correct, so do not forget to include it.

Since the DL405 CPUs perform math operations in BCD, it is usually best to convert the data to BCD immediately. You can leave out this instruction if your application does not require it (such as for PID loops, which require the process variable to be in binary format).

This load instruction reads the data into the accumulator again. The channel data will be pushed into the first level of the stack.

This instruction masks the analog data values and leaves the channel ID bits in the accumulator.

Now you have to shift the accumulator bits so the channel ID bits will result in a value between 0 and 15 (binary format).

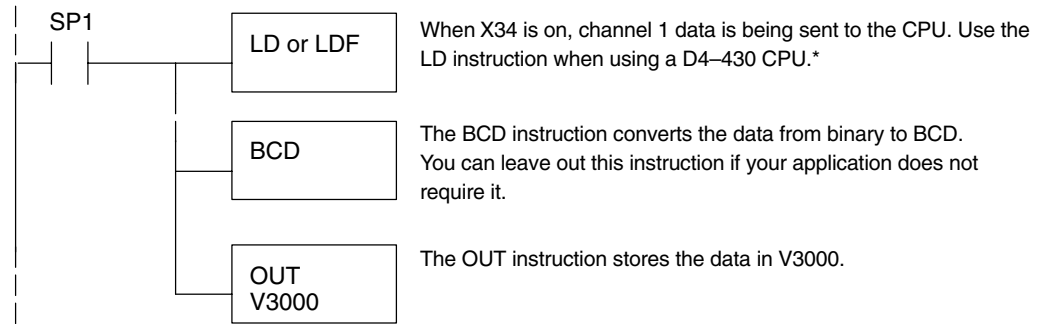
OUTX copies the value from the first level of the accumulator stack to a source address offset by the value in the accumulator. In this case it adds the above binary value (0-15) to V3000. The particular channel data is then stored in its respective location: For example, if the binary value of the channel select bits is 0, then channel 1 data is stored in V-memory location V3000 (V3000 + 0) and if the binary value is 6, then the channel 7 data is stored in location V3006 (V3000 + 6). See the table on page 7-13.

Note, this example uses SP1, which is always on. You could also use an X, C, etc. permissive contact.

Single Channel Selected

✓ ✓ ✓
430 440 450

Since you do not have to determine which channel is selected, the single channel program is even more simple.



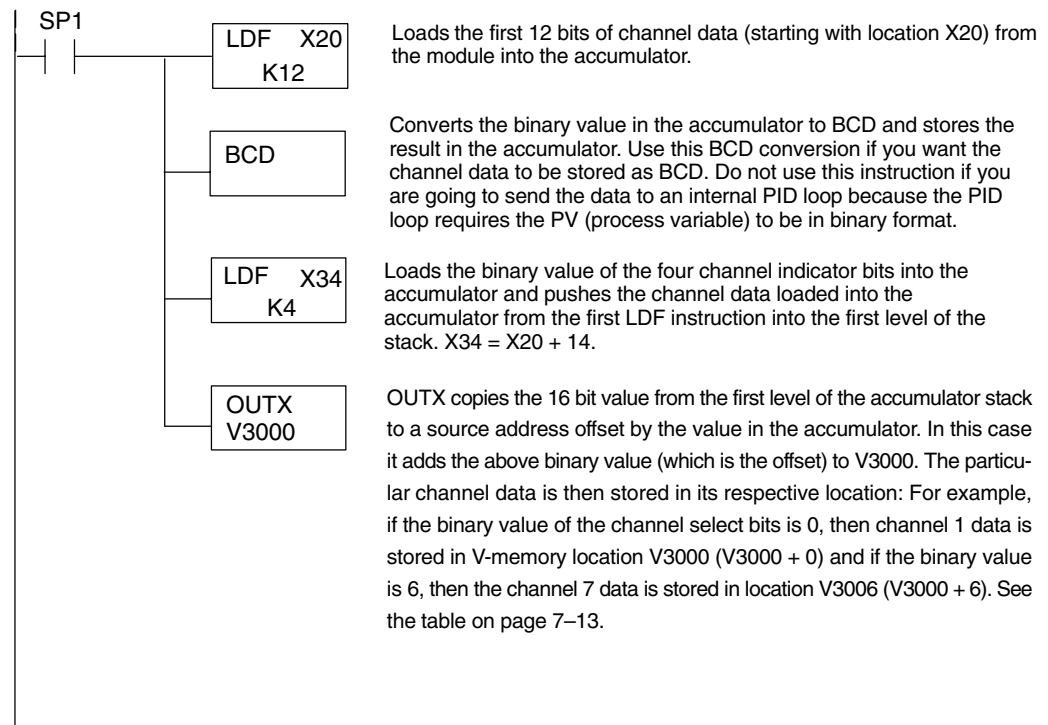
Note: This example uses SP1, which is always on. You could also use an X, C, etc. permissive contact.

* Remember, before the BCD instruction is executed, the D4-430 requires an additional instruction to mask out the first four bits that are brought in with the LD instruction. An example of how to do this using an ANDD instruction is shown in the previous section.

Reading Values

✗ ✓ ✓
430 440 450

The following program example shows how to read the analog data into V-memory locations with D4-440 and D4-450 CPUs. Once the data is in V-memory, you can perform math on the data, compare the data against preset values, and so forth. This example will read one channel per scan, so it will take sixteen scans to read all sixteen channels.



Note: This example uses SP1, which is always on. You could also use an X, C, etc. permissive contact.

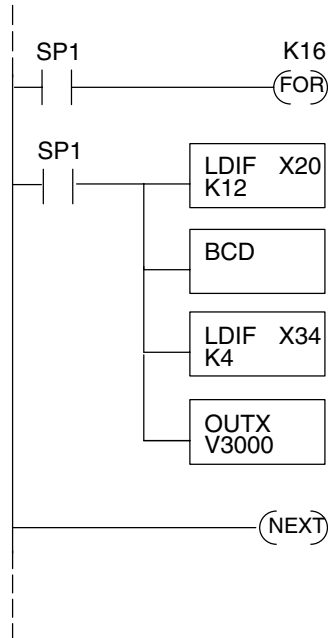
Reading Sixteen Channels in One Scan

430 440 450



The following program example shows how to read all sixteen channels in one scan by using a FOR/NEXT loop. Before you choose this method, do consider its impact on CPU scan time. The FOR/NEXT routine shown here will add about 32ms (2ms/loop) to the overall scan time. If you do not need to read the analog data on every scan, change SP1 to a permissive contact (such as an X input, CR, or stage bit) to only enable the FOR/NEXT loop when it is required.

NOTE: Do not use this FOR/NEXT loop program to read the module in a remote/slave arrangement; it will not work. Use one of the programs shown that reads one channel per scan.



Starts the FOR/NEXT loop. The constant (K16) specifies how many times the loop will execute. Enter a constant equal to the number of channels you are using. For example, enter K4 if you're using 4 channels.

Immediately loads the first 12 bits of the data word (starting with X20) into the accumulator. The LDIF instruction will retrieve the I/O points without waiting on the CPU to finish the scan.

Since the DL405 CPUs perform math operations in BCD, it is usually best to convert the data to BCD immediately. You can leave out this instruction if your application does not require it (such as PID loops).

This LDIF instruction immediately loads the four channel indicator bits into the accumulator.

The OUTX instruction stores the channel data to an address that starts at V3000 plus the channel offset. For example, if channel 2 was being read, the data would be stored in V3002 (V3000 + 2). See the table on page 7-13.

Note, this example uses SP1, which is always on. You could also use an X, C, etc. permissive contact.

Scaling the Input Data

Most applications usually require measurements in engineering units, which provide more meaningful data. This is accomplished by using the conversion formula shown.

You may have to make adjustments to the formula depending on the scale you choose for the engineering units.

$$\text{Units} = A \frac{H - L}{4095}$$

H = high limit of the Engineering unit range

L = low limit of the Engineering unit range

A = Analog value (0 – 4095)

For example, if you wanted to measure pressure (PSI) from 0.0 to 99.9 then you would have to multiply the analog value by 10 in order to imply a decimal place when you view the value with the programming software or a handheld programmer. Notice how the calculations differ when you use the multiplier.

Analog Value of 2024, slightly less than half scale, should yield 49.4 PSI

Example without multiplier

$$\text{Units} = A \frac{H - L}{4095}$$

$$\text{Units} = 2024 \frac{100 - 0}{4095}$$

$$\text{Units} = 49$$

Handheld Display

```
V 3101 V 3100
V MON 0000 0049
```

Example with multiplier

$$\text{Units} = 10 A \frac{H - L}{4095}$$

$$\text{Units} = 20240 \frac{100 - 0}{4095}$$

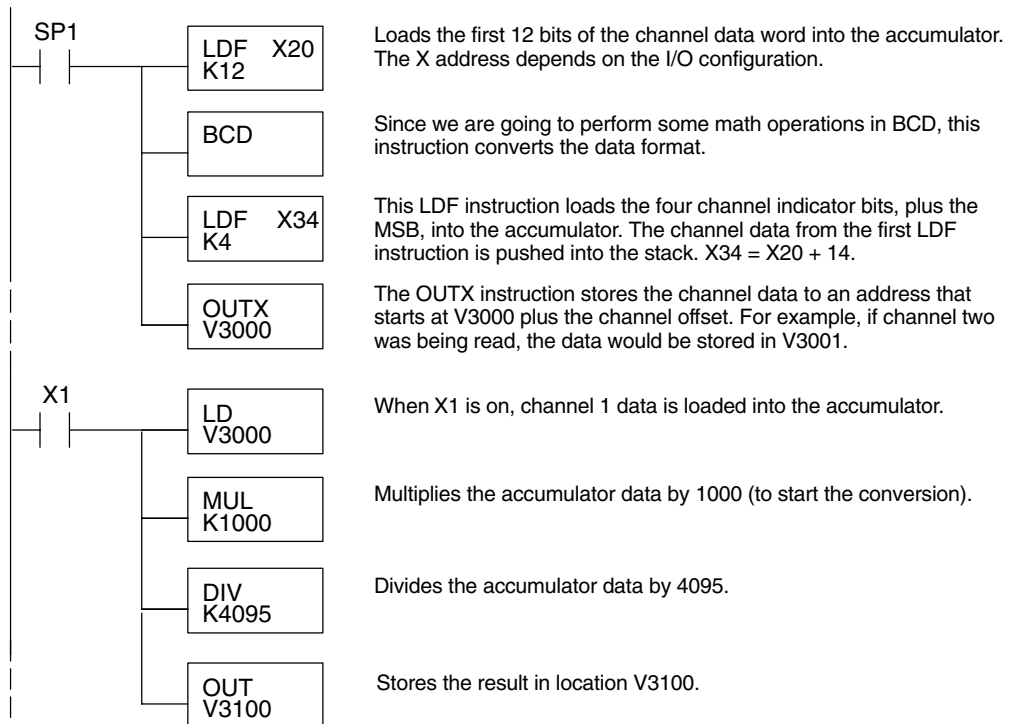
$$\text{Units} = 494$$

Handheld Display

```
V 3101 V 3100
V MON 0000 0494*
```

*Value is more accurate

Here's how you would write the program to perform the engineering unit conversion. This example uses SP1 which is always on. You could also use an X, C, etc. permissive contact.



Analog and Digital Value Conversions

Sometimes it is helpful to be able to quickly convert between the signal levels and the digital values. This is especially useful during machine startup or troubleshooting. The following table provides formulas to make this conversion easier.

Range	If you know the digital value ...	If you know the signal level ...
0 to 5V	$A = \frac{5D}{4095}$	$D = \frac{4095}{5} (A)$
0 to 10V	$A = \frac{10D}{4095}$	$D = \frac{4095}{10} (A)$

For example, if you are using the 0V to +10V range and you have measured the signal at 6V, you would use the following formula to determine the digital value that should be stored in the V-memory location that contains the data.

$$D = \frac{4095}{10} (A)$$

$$D = \frac{4095}{10} (6V)$$

$$D = (409.5) (6)$$

$$D = 2457$$